**Octojack.**

**Background**

**Overview of Octojack.** Octojack (<https://bitbucket.org/natecermak/octojack>) is a breakout board for the Teensy 3.6. With a fully-assembled octojack, you can program the Teensy 3.6 microcontroller in order to do the following:

turn stepper motors

set the position on servo motors

turn DC motors

turn LEDs on and off

toggle power to devices on or off (+5V or +12V)

receive and record analog or digital voltages (0-3.3V)

generate digital or analog voltages (0-3.3V)

**Microcontroller.** Octojack is designed for use with Teensy microcontrollers (<https://www.pjrc.com/teensy/>), specifically Teensy 3.6, but should work fine with Teensy 3.5. Many features will not work with a Teensy 4.0 or 3.2 – do not use these.

**Power.** By default (and as long as J33 is not jumpered), the Teensy is powered over USB, and hence doesn’t depend on whether the +5V or +12V power supplies are plugged into the Octojack. The USB power supply (5V) can be used as the +5V power supply for the board by jumpering J33 – however, you should absolutely NOT do this while a separate +5V supply is plugged into the board. In general, the USB power supply will only be good for light loads, so jumpering J33 is typically not a great idea.

The stepper motor drivers, DC motor drivers, and relays can be selected to be driven from either the +5V line or the +12V line, using the jumpers on the back of the board. The servo motors are driven exclusively from the +5V line, and the LED drivers (Picobucks) run exclusively on the +12V line.

If the appropriate power supply is not plugged in, then modules relying on that power supply definitely will not work.

**Building**

**Ordering parts.** You will need to purchase [about $150 worth of components](https://bitbucket.org/natecermak/octojack/src/1078f867cc0ab1765c5eccc8103f68c636fa0186/octojack_BOM.xlsx) from [Digikey](https://www.digikey.com/) and a [printed circuit board (PCB)](https://bitbucket.org/natecermak/octojack/src/1078f867cc0ab1765c5eccc8103f68c636fa0186/octojack_fabFiles_mainBoard.zip) and [panels for the box](https://bitbucket.org/natecermak/octojack/src/1078f867cc0ab1765c5eccc8103f68c636fa0186/octojack_fabFiles_panel.zip) from any of a number of companies - I often use [JLCPCB](https://jlcpcb.com/) or [seeedstudio](https://www.seeedstudio.com/fusion_pcb.html). To order the main board, upload the [main board gerber files](https://bitbucket.org/natecermak/octojack/src/1078f867cc0ab1765c5eccc8103f68c636fa0186/octojack_fabFiles_mainBoard.zip) to your PCB fabrication site of choice. Most of the default options should be fine (FR-4, 1.6mm thick, any color is fine, etc), but you'll need to select a 4-layer board, and if you need to enter the board dimensions, the main board is 159 mm by 79.5 mm in size. To order the panels (the front and back of the box), upload the [panel gerber files](https://bitbucket.org/natecermak/octojack/src/1078f867cc0ab1765c5eccc8103f68c636fa0186/octojack_fabFiles_panel.zip) to your PCB fabrication site of choice. Select 2-layer board, size 170 mm by 108.5 mm. At least with [JLCPCB](https://jlcpcb.com/) and [seeedstudio](https://www.seeedstudio.com/fusion_pcb.html), you also need to enter that there are two designs in this file. Typically I order 5 of each board, since the cost is essentially the same as ordering 1 board.

**Assembly.** Once you've got the main board, panels and [components](https://bitbucket.org/natecermak/octojack/src/1078f867cc0ab1765c5eccc8103f68c636fa0186/octojack_BOM.xlsx), you you'll have to do some soldering. I suggest you have the following handy:

* soldering iron
* solder paste
* solder wire
* tweezers
* safety glasses

I suggest soldering components in the following order, as some components occlude access to others.

1. Start with all the surface mount components.
2. Pushbutton (SW1). Before soldering, cut off about 2mm from each leg so that the pins are nearly flush with the board
3. Right angle 3x4 pin header (U3) on back side of board and C8 (front side).
4. Teensy female pin headers (2 24-pin female headers).
5. 2-pin headers (J11-15, J17, J24 and J28, J33) on the back side of the board!
6. Push terminals (7 four-position and 1 six-position), but first cut off about 2mm from each leg so that pins barely protrude from the other side of the board.
7. All right angle connectors EXCEPT J4 on the back side of the board!
8. 2x3-pin headers for J7, J18, J27, and 3-pin headers for J16,J21, and J30, all on the back side of the board!
9. 8-pin female pin header sockets for A1 A2 and A3
10. capacitors C1-3.
11. 3-pin headers J3, J5, J6 and J32 on the back of the board!
12. 8-pin female pin header sockets for U2
13. Power plugs (J1, J2)
14. Eight BNC connectors for GPIO.

Next, attach jumpers to select the desired power source (5V or 12V) for each output (relays, DC motors, and stepper motors). Also attach jumpers to select the desired microstepping mode for the stepper motors (using the DRV8825 datasheet).

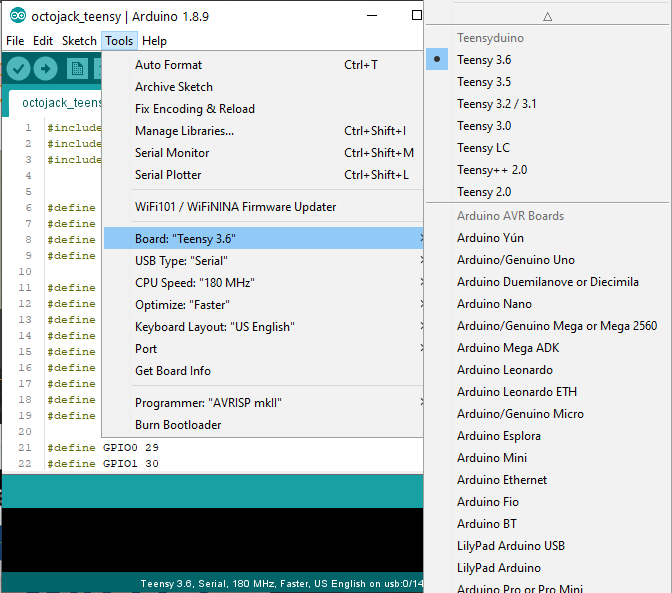
Attach the Teensy, TB6612FNG breakout board, and DRV8825 modules by gently pressing them into their sockets. If needed, set the current limits for the DRV8825 stepper motor driver modules.

Finally, place the entire board inside the enclosure (middle slot), attach the panels, and plug in the USB.

**Programming**

**Needed software.** Octojack can be programmed via the Arduino IDE (<https://www.arduino.cc/en/main/software>). After installing the Arduino IDE, you will additionally need to install Teensyduino (<https://www.pjrc.com/teensy/td_download.html>). In the Arduino IDE, you can write C or C++ programs (called “sketches”), which can be compiled and run on the Teensy (among many microcontrollers). The IDE itself includes a number of examples (<https://www.arduino.cc/en/Tutorial/BuiltInExamples>), which are likely to be helpful getting started.

**Compilation.** To compile a program, press the checkmark in the upper left hand corner of the window. At the bottom of the window, you will see any error messages that arise. To upload your program to the Teensy, first make sure the Teensy is connected to the computer via a USB cable. Next, make sure the board and port are set correctly, under the menu bar Tools > Board (should be set to Teensy 3.6) and Tools > Port. You can figure out which port corresponds to your Teensy by plugging and unplugging the Teensy into the computer and seeing which port disappears and reappears. Finally, click the right-arrow button next to the checkmark to upload your program. If all goes well, a small Teensyduino window should open, a progress bar should complete, and it should show the message “Reboot OK”.



Once the Teensy has been programmed, the program stays on the Teensy even if the power is turned off. You can change the program running on the Teensy by uploading a new program.

If you are comfortable with basic C programming and the standard Arduino and Teensy libraries, you can use a basic template sketch (<https://bitbucket.org/natecermak/octojack/src/master/octojack_teensy/octojack_teensy.ino>) that includes definitions of all the pins and a few helpful objects (AccelStepper, Servo) to control them.

**Handreach task – and installation of LABVIEW**

**Installing LabVIEW and dependencies** (note that this can take a few hours).

1. Install Labview 2018 (<https://www.ni.com/en-us/support/downloads/software-products/download.labview.html#306369>). Driver support is not necessary. **Version 2018-SP.  
license is via these steps :** <https://cis.technion.ac.il/files/Labview_Installation_Network_License.pdf>.

2. Install the **Vision Development Module** (<https://www.ni.com/en-il/support/downloads/software-products/download.vision-development-module.html#305253>). **This version in now 18.5 (default).**

3. Install the vision acquisition software (<https://www.ni.com/en-il/support/downloads/drivers/download.vision-acquisition-software.html#306476>). **Version 18.5.1** (in accordance to **Vision Development Module**).

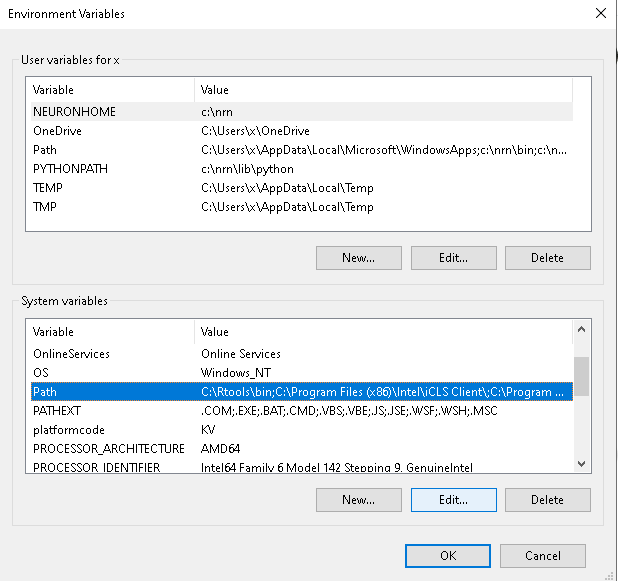
4. Install NI-DAQmx driver (<https://www.ni.com/en-il/support/downloads/drivers/download.ni-daqmx.html#325032>). **Version 19.5**.  
**\*\*\* you can also install it via NI Package Manager and also see all the drivers and modules.**

5. Install ffmpeg (<https://www.ffmpeg.org/>). Labview’s built in codecs suck, so ffmpeg is used for H.264 compression of the behavioral videos.

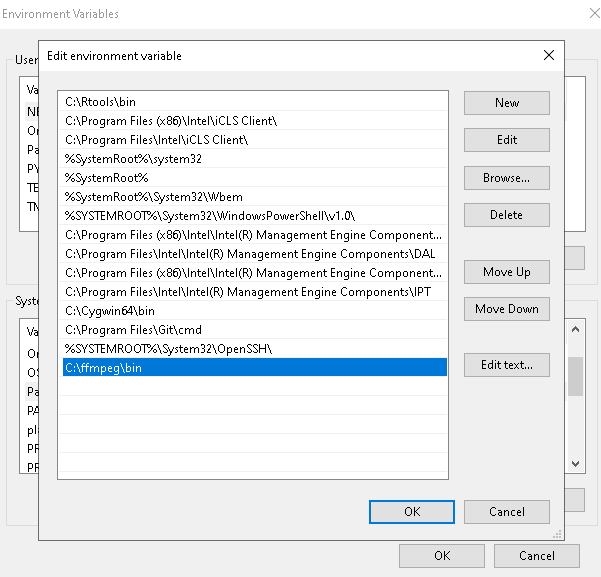
# see if ffmpeg is installed via cd (command window) – if it recognize, stop here – all good 

If not ("ffmpeg is not recognized as an internal or external command")… do the following for adding the ffmpeg's 'bin' folder to your file path:

click the start menu icon, type "path", select "edit the system environmental variables". click "environment variables" at the bottom of the window, then find the variable "Path" in the bottom half.



click "edit" and add your ffmpeg/bin directory, wherever that might be.



hit okay a bunch of times to close these windows, then restart command prompt, and make sure that you when you type ffmpeg at the command prompt, you get output from ffmepg!

**Notes:** The program will likely work fine with later versions of Labview, but (1) make sure that the versions of Labview, Vision Development Module, and Vision Acquisition Software are all the same (2018!), and (2) if you save a program in a later version of Labview, it won’t work if you try to run it on an earlier version.

**Code.** All code for the handreach task (and treadmill recordings too) lives at <https://bitbucket.org/natecermak/mousetaskcontroller>. You can download the code directly from there, or install git (<https://git-scm.com/downloads>) and then clone the repository using the command

git clone <https://bitbucket.org/natecermak/mousetaskcontroller>

from the command line.

**Programming the Octojack.** As discussed above in the Octojack programming section, upload the file “teensy2pTriggerMediator.ino” to the Octojack.

**Starting the handreach task.** Double-click TwoPhotonTrialController\_reachTask.vi, or open LabView and then open this file. Click the arrow in the upper left corner of the window to start the program.

**Physical connections for the Octojack for the handreach task.**

**Errors.** Labview may throw several kinds of errors. Here are some debugging tips.

* If the error message includes the word “VISA”, it has something to do with a serial connection, likely the Octojack. Make sure the Octojack COM port is set correctly (on the “Don’t touch” tab).
* If the error message says anything about TCP, it likely failed to connect to PrairieView. Is PrairieView running? Does the Labview VI have the correct IP address for the 2P computer? (also on the “Don’t touch” tab.)
* If it says something about “unknown error” - close Labview entirely, then re-run Labview.